Exploring Information - Implementing an Entity-based Search Engine

Anna Luhtakanta

Helsinki June 10, 2019

Master's Thesis

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author

Anna Luhtakanta

Työn nimi — Arbetets titel — Title

Exploring Information - Implementing an Entity-based Search Engine

Oppiaine — Läroämne — Subject

Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's Thesis	June 10, 2019	32

Tiivistelmä — Referat — Abstract

Finding and exploring relevant information from a huge amount of available information is crucial in today's world. The information need can be a specific and precise search or a broad exploratory search, or even something between the two. Therefore, an entity-based search engine could provide a solution for combining these two search goals.

The focus in this study is to 1) study previous research articles on different approaches for entity-based information retrieval and 2) implement a system which tries to provide a solution for both information need and exploratory information search, regardless of whether the search was made by using basic free form query or query with multiple entities. It is essential to improve search engines to support different types of information need in the incessantly expanding information space.

Avainsanat — Nyckelord — Keywords

Entity Search, Exploratory Search, ElasticSearch, Entity Linking, Query Expansion,

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

Searching for information in today's world is daily life for everyone. Finding relevant information from the mass of data we have now is becoming harder and harder. Also, the enormous amount of data makes it hard for users to explore and express their information needs (Ruotsalo, Jacucci, Myllymäki, & Kaski, 2014). Although the information need varies for the user, structuring queries, which represents this need of information, grows more complicated together with the complexity. The basic search box custom doesn't serve these more complex queries, where the scope can be from keywords to a typical natural language question (Sawant & Chakrabarti, 2013; Balog, 2018).

The information need or information search behavior (Athukorala, Głowacka, Jacucci, Oulasvirta, & Vreekenc, 2015; Kim, 2009) can be divided into 3 categories, where a user has 1) a specific and precise search goal, 2) an interpretive search goal, or 3) a broad exploratory search goal. To satisfy all of these search goals and complex queries, information retrieval should concentrate more on entities themselves instead of just finding documents where they are mentioned (Balog, Meij, & de Rijke, 2010). Looking up for one specific entity can lead to new interesting information when the system provides more entities related to the target and makes it easier to browse and navigate in the information space (Klouche, Ruotsalo, & Jacucci, 2018).

Entities and Entity Search are rapidly growing in popularity on the research fields of Information Retrieval and Semantic Search. Nevertheless, previous works haven't fully been able to provide semantically, contextually and non-obviously related entities for vague search terms and queries. In this thesis, we want to approach this problem from an aspect of exploratory search.

Designing IR systems for exploratory search has been researched a lot, but there are still improvements to be made. This study suggests that entity search can add more value for the user when combining with an exploratory search. Especially when thinking that the search engine contains usually two building blocks: the retrieval system and the user interface.

Visual user interfaces can allow users more control to perform search and improve the discovery of more novel information (Metzger, Schenkel, & Sydow, 2013; Ruotsalo et al., 2014). These have been seen in retrieval tasks with too complicated or too vague queries. To implement a search engine, which can satisfy the information need of all kind, the retrieval system and the user interface should complement each other.

1.1 Objectives and Scope

The implementation described in this thesis is part of a research project at the University of Helsinki. The purpose of the research is to provide a novel search engine aimed at supporting exploration of innovation at the University

of Helsinki through the use of interactive visualizations and entity recommendation.

Within that project, I have been tasked with designing and implementing a retrieval system indented to assist a visual entity-based exploratory search engine for innovation.

The purpose of this study is to improve task performance and to increase the quality of relevant information by utilizing entity linking. The evaluation will be performed by measuring Precision and Normalized Discounted Cumulative Gain and comparing the search results with the state-of-the-art TUHAT search engine.

The thesis focuses on the entity retrieval system. The data used in this project comes from the TUHAT database, a research portal of the University of Helsinki. The data has been studied and modified to be suitable for entity-based structure.

Visualization of the search engine is not related to my personal contribution and therefore the user interface has been considered beyond the scope of this thesis. We are keeping mind the user interface of this project, considering that it will exploit the implementation designed and created in this thesis.

1.2 Research Questions

RQ1 - Does entity linking improve information exploration comparing to basic query search, especially when the search is made with multiple entities?

The first research question asks if the search engine should use entity linking while processing queries, particularly in cases where query contains more than one entity.

RQ2 - How to retrieve relevant and diverse results from different entity types?

The second research question asks if we manage to design and implement a system, which utilizes different types of entities and retrieves heterogeneous result sets. In other words, we want relevance and variety from our result sets which ease the information exploratory point of view.

1.3 Chapters

This thesis is structured as follows: Chapter 2 provides the background, describes relevant terms and related work. Chapter 3 lists a few of the problems related to the design of entity-based IR systems and introduces the intended user interface and how the retrieved information will be visualized. Chapter 4 goes through the structure of the implemented search engine. The evaluation of the system and its results can be found in chapter 5. And in chapter 6 there is the discussion and future work. And finally, in chapter 7 will end this thesis

with the conclusion.

2 Background

In this chapter, we explain all the related terms and review previous works related to information retrieval, exploratory search and entity search.

2.1 Information Retrieval

> "Information Retrieval is a field concerned with the structure, analysis, organization, storage, searching and retrieval of information."
>
> *Salton (1968)*

Information retrieval (referred also as *IR*) has been a highly researched area because of the rapid increase of the knowledge and popularity of the internet and search engines. Information retrieval includes different tasks related to information (Croft, Metzler, & Strohman, 2009). In this study, we focus on retrieving information based on queries, and leave other IR related tasks, like crawling and indexing, out.

Information - has different kinds of forms depending on the context. Among many things, it can be connected to communication, knowledge or data. In this study, the word "information" relates more to its data aspect. Information - or in this case data can have a *structured*, *semi-structured* or *unstructured* format (Balog, 2018; Liu, Fang, Chen, & Wang, 2012). Structured data has a strict predetermined schema of how the information is organized, and all those attributes have to be present, similar to a relational database. Example: Book with title, author, date, and publisher. Opposite of that is unstructured data. Like web pages, there is no predefined format and the content can be any kind of information. The semi-structured data is somewhere between these two. It doesn't demand that all the attributes are present. The best example of this is a JSON object.

Search - is usually perceived as a task triggered by an information need and it can be conducted in many ways (Croft et al., 2009; Büttcher, Clarke, & Cormack, 2016; Kim, 2009). Such as *web search*, *peer-to-peer search*, *vertical search*, *desktop search* and *entity search*.

Earlier IR systems have been concentrated on documents and Document Retrieval. This is shifting towards entities and Entity Retrieval, and in many cases, the same methods and schemes in document retrieval can be used for entities (Balog, 2018).

There are three main retrieval models used in IR systems: Boolean, Vector, Probabilistic (Baeza-Yates & Ribeiro-Neto, 1999; Croft et al., 2009). The

Boolean model uses set theory and Boolean expressions to retrieve documents. The Vector model uses vectors to retrieve and rank documents based on the query. The Probabilistic model retrieves and ranks documents based on their probability of relevance. Many studies mentioned in this thesis are using Vector Space or Probabilistic model, like Language models or BM35, in their work. There are also increasing usage of machine learning in IR models, such as Learning-to-rank approaches.

2.2 Information Need

Information need is used to implicate user's reason for searching information and using search engines (Kim, 2009; Büttcher et al., 2016). This need can be divided into 3 categories, where the user has 1) a specific and precise e.g. lookup search goal, 2) an interpretive search goal, or 3) a broad exploratory search goal (Athukorala et al., 2015; Kim, 2009).

In **The lookup task**, the user already has a closed question to look for a specific answer. "When is Finland's Independence Day?" is a lookup task to find the date for celebrating Finland's Independence and the user already expects the answer to be a date. This type of task is easy to express, thus generating a query for it, is something the user can do without a problem (Ruotsalo et al., 2014).

The interpretive task has, on the other hand, more an open-ended question. The goal is somehow known, but the answers can be more than one. "What kind of food is Finnish traditional food?" The user is focusing on the traditional food in Finland and might have some idea what kind of food there is. Still, this might return different results e.g. based on the province, which the user didn't anticipate (Kim, 2009).

The exploratory task is more about investigating and learning of a topic and widening the knowledge on it. There are no specific goals or boundaries which could indicate in advance that the search has been completed. In many cases, the search also expands and evolves to other unknown topics. In these cases, the user usually experiences hardships on creating queries for the need (Ruotsalo et al., 2014). Another aspect of exploratory task relates to imprecise queries when the user doesn't know terms, or the query structure search engine uses. The opposite for this is Expert search, where the user looking for information is used to build queries (Hasibi, Balog, & Bratsberg, 2016; Bron, Balog, & de Rijke, 2010; Balog, 2018). An example of an exploratory task could happen when the user is moving to Finland and wants to know more about "Finnish law and immigration".

2.3 Exploratory Search

As mentioned in the previous section, in exploratory search the information need is still uncertain, and the user has to do complex or multiple queries to navigate in the search space. Designing IR systems for exploratory search has been researched a lot, but there are still improvements to be made. Understanding and expressing the information need as queries can be hard for the users, because of the uncertainty and expanding area of topics (Ruotsalo et al., 2014; Athukorala et al., 2015). Search engines are mostly build to handle simple queries for lookup tasks and the nature of the exploratory search can be hard to perform in those kinds of systems (Croft et al., 2009; Athukorala et al., 2015; Balog, 2018).

> *"Users need search engines and user interfaces that adapt to their capabilities and search behavior, rather than require them to adapt to them."* — Ruotsalo et al. (2014)

Exploratory search tries to solve problems related to:

- **Limitation on the initial result set** (Ruotsalo et al., 2014), where the search engine should present a wider set of results, including additional highly related topics. With this, we can assure exploration beyond the initial query.
- **Recommendations for more diverse results** (Metzger et al., 2013; Klouche et al., 2018), where the search engine should present recommendations based on given examples. E.g. while searching movies, the system should recommend similar movies, authors, directors and so on, which might give more valuable information for the user to explore.
- In our case we also include **Serendipitous Search** (Bordino, Mejova, & Lalmas, 2013) for the exploratory search, considering that relevant, interesting and novel information is something we want for our system to provide.

Our intention is to implement a retrieval system which could be exploited by the interface suggested by Klouche et al. (2018). There have been done other similar work, where the visualization of the data and user's ability to interact with it, improves the exploratory search (ExplorationWall (Klouche et al., 2015), RelevanceMap (Klouche, Ruotsalo, Micallef, Andolina, & Jacucci, 2017) and SciNet/Intent Radar (Ruotsalo et al., 2014, 2013)).

In the study of ExplorationWall, they used the personalized PageRank to rank entities and ranking documents was based on the unigram language model and maximum likelihood with Jelinek-Mercer smoothing to rank documents.

Klouche et al. (2017) proposed an interactive visualization technique for multi-aspect information retrieval, called RelevanceMap. They used vector-space and multinomial unigram language modeling with Bayesian Dirichlet smoothing and for the re-ranking probability ranking principles combined with the query phrases got from the visualization.

In SciNet (Ruotsalo et al., 2014, 2013), they used information visualization and interactive user modeling with machine learning, to help users to explore around the information space.

Bordino et al. (2013) also addressed a similar problem and used Vector-Space modeling and Personalized PageRank in their work to obtain interesting and surprising information.

2.4 Entity

A term **entity** can be inspected from different angles. In Wiktionary [1] it has 4 types of meanings:

- An entity has a distinct existence as **an individual unit**. Often used for organizations which have no physical form.
- The **existence** of something considered **apart from its properties**.
- (databases) Anything about which **information** or data can be stored in a database; in particular, an organized array or set of **individual elements** or parts.
- The state or quality of being or **existence**.

All of them have something to do with a unique existence. Balog (2018) interprets that the term "Entity" is a real-world object with a unique identifier and relationships to other objects. An entity usually has names and other attributes and can be categorized by type. When an entity is given as a summary of it with key elements, it's often called *Entity Card* The most well-known entity types are people, locations and organizations.

Wikipedia[3] is one of the most well-known entity-based data collections, where a single entity has its own page with relevant information and hyperlinks to other entities (Shen, Wang, & Han, 2015). These hyperlinks and related entities, as seen in Table 1, can be model as nodes and edges which form a graph (Klouche et al., 2018). These are known as *Knowledge Graphs* which are stored in *Knowledge Base* There are various knowledge bases, like DBpedia, Freebase, and YAGO. Wikipedia is considered to be more as a knowledge repository than as a knowledge base (Balog, 2018) due to its structure.

[1] https://en.wiktionary.org/wiki/entity
[2] https://www.imdb.com/title/tt5884052
[3] wwww.wikipedia.com

Pokémon Detective Pikachu			
Name	*Pokémon Detective Pikachu*	**Genres**	*Action, Adventure, Comedy, Sci-Fi*
Aliases	*Detective Pikachu, Pokemon*	**Type**	*Movie*
Director	*Rob Letterman*	**Release Date**	May 3 2019
Starts	*Ryan Reynolds, Justice Smith, Kathryn Newton...*	**Description**	"The story begins when ace detective *Harry Goodman...*"

Table 1: One example of an entity and entity card, a movie from IMDb[2]. Italicized text symbolizes entities

2.5 Entity Search

Entity search or *Entity-oriented search*, focuses on returning a result set of entities related to a given source entity or an unstructured query (Metzger et al., 2013; Balog et al., 2010; Balog, 2018). Entities on the queries for the retrieval can be represented as same as terms and perform traditional retrieval tasks on them (Hasibi et al., 2016; Balog, 2018).

It's important that the related entities are semantically meaningful. Entity retrieval can be interpreted as *Entity ranking* (Bron et al., 2010; Sawant & Chakrabarti, 2013), *List completion* (Metzger et al., 2013) or *Related entity finding* (Li, Li, & Yu, 2010; Balog et al., 2010). Entity ranking refers to a ranked list on a specific entity category. List completion concentrates on similar entities based on the initial query and entities. Related entity finding returns entities based on the source entity with wanted relation and category. This study is focused more on the list completion but has traits also from the related entity finding.

Recent studies have used different approaches for entity search from traditional document retrieval techniques, like language models or BM25 (Balog, 2018) and semi-structured retrieval models (Hasibi et al., 2016), to learning-to-rank approaches (Sawant & Chakrabarti, 2013; Xiong, Liu, Callan, & Hovy, 2017). For example, Markov Random Field model (Hasibi et al., 2016), KL-divergence retrieval model (Gottipati & Jiang, 2011), learning-to-rank model with BM25, TF-IDF, Coordinate Match, and language model with Dirichlet smoothing (Xiong et al., 2017) and Bron et al. (2010) built their entity retrieval framework from co-occurrence models, type filtering, context modeling, and homepage finding.

In this study, the approach will be based on unsupervised ranking methods (BM25) instead of supervised ranking methods (e.g. learning-to-rank)

Some approaches focus on retrieving related entities based on the type (Vallet & Zaragoza, 2008). But instead of categorizing and predefining types

of entities used in retrieval, we provide all the relevant entities regardless of the type. We want to give the most divergent result sets which might interest the user.

To provide entities that are interesting in terms of related topics, new keywords, researchers on that field, articles with the same authors or topics. For example, if we are looking for information about a researcher, interesting information would be the articles he wrote, the projects he was in, the area and topics he researches, and colleagues he is working with or other researchers who are working on the same topics.

2.6 Entity Linking

Entity linking is used for attaching entities with relationships together in a knowledge base. Like a company has its workers, location, and products. These all are linked together and can be used for exploratory search, because of the additional information it can provide around the entity (Klouche et al., 2018). For instance, providing background information or recommending related entities. With entity linking, information retrieval performance can be improved on finding semantically related entities (Shen et al., 2015; Dalton, Dietz, & Allan, 2014).

The usage of entity linking has previously been in finding contextual and semantic similarities between a document and candidate entities in long texts. Recently also short texts, like queries and tweets, have been the focus in entity linking research (Hasibi et al., 2016).

The term *Entity Linking* can also be used to represent other tasks, like populating or merging knowledge bases or question answering. In this study, we are more interested in the usage of entity linking in queries, as mentioned in the article written by Shen et al. (2015).

The implementation in this study will also exploit the query expansion technique.

3 Problem

In this chapter are listed some of the problems we are interested in which are related to designing an entity-based IR system. There are also use case scenarios of 1) a single search, 2) a multi-entity search, and 3) showing an entity card, and how those would look on the intended user interface of this search engine (Klouche et al., 2018). This interface should be able to exploit the implemented IR system.

Query processing - to create a search engine that can process two types of queries: single entity and multi-entity queries, and retrieve comprehensive information of regardless the entity types and attributes. Creating queries that would fit concurrently for both structured and unstructured attributes is not as simple as just searching from a single structured attribute.

Relevance - to find the entities which have the information that satisfies the user's information need. There are different factors that make information relevant.

Vocabulary mismatch problem and ambiguity (Croft et al., 2009; Gottipati & Jiang, 2011) - to provide relevant documents regardless of different kind of grammar points, stemming or ambiguity of the entity names. The system should understand to combine singles and plurals (like Query - Queries) and different aliases for search queries (like Data mining - Information harvesting) and broad for related topics (like Data mining - Machine learning). Also, the representations of the queries should maintain the same word order as phrases, since two or more words meaning can be semantically different if together or separately.

Single entity search

Figure 1: First step. User can conduct a search based on a keyword or an entity (Klouche et al., 2018).

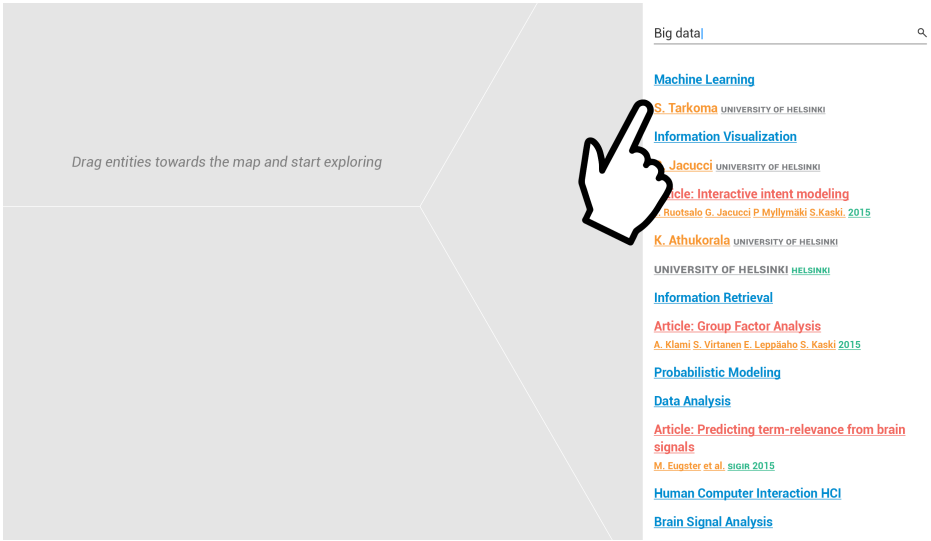

Figure 2: Second step. User can see the result list

Single entity search: User starts a search by typing a keyword or a named entity (Fig.1). The interface will update the screen and shows the results based on the given query from the input field (Fig.2). As in the scenario, the user wants to know about *big data*. After typing the keyword "Big Data", the user finds entities, which are related to big data. Such as researchers, articles, projects and other keywords associated with big data.

Multi-entity search

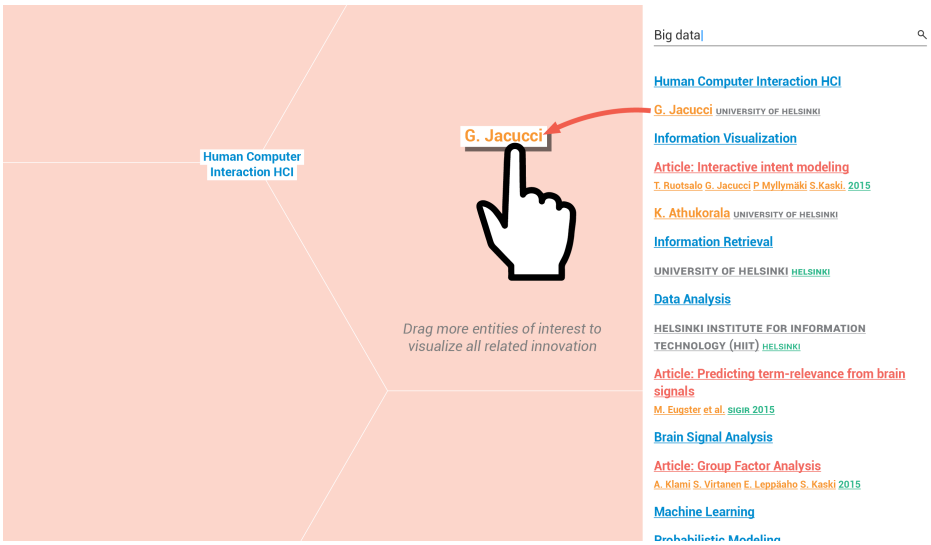

Figure 3: Third step. User can select entities from the result list and explore the information space

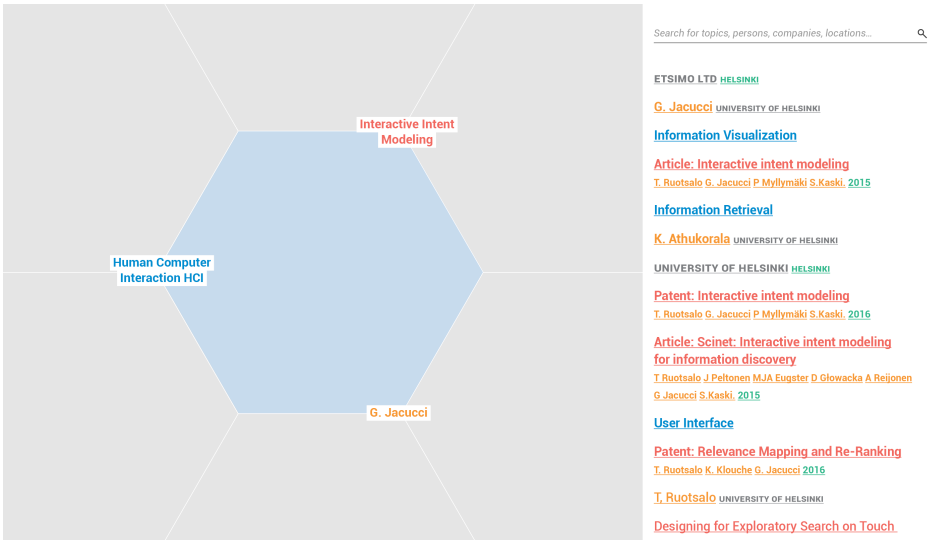

Figure 4: Fourth step. User has selected different entities, regardless of the type or category and can see the updated result list

Multi-entity search: User can drag entities on the interface (Fig.3) and do a multi-entity search based on the dragged entities. The list of entities on the right will be updated with the result set based on the selected entities on the search area and the user can continue exploration (Fig.4). Now the user has selected and dragged different types of entities, *Human-Computer Interaction, Interactive Intent Modeling*, and *Giulio Jacucci*, for the search. User can discover more entities which are similar to these three entities.

Detailed entity card

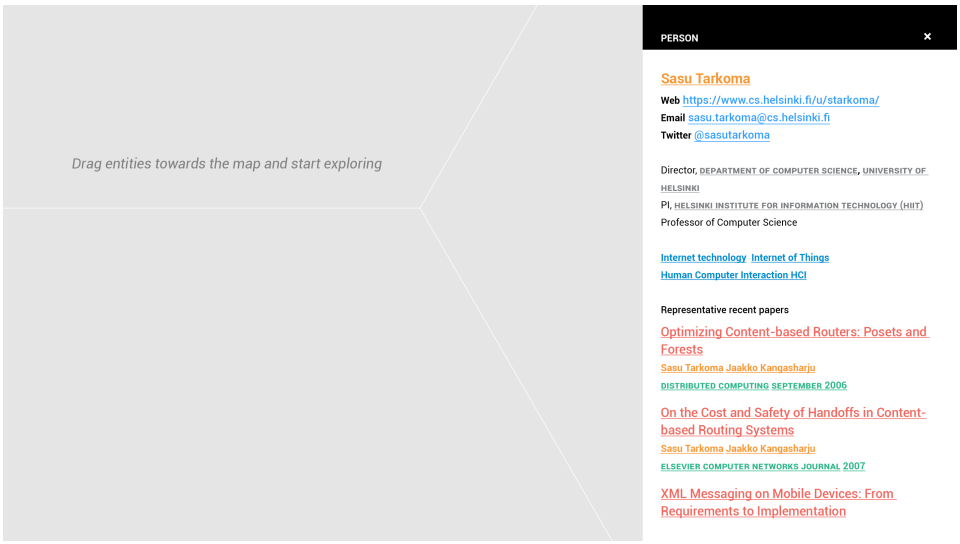

Figure 5: Fifth step. User can look more information of a specific entity

The user can explore the returned result list (Fig 2) and also look up more detailed information of an entity (Fig 5). This entity card will provide more additional information and entities related to it. Here in the example, then user has selected *Sasu Tarkoma* and can see the entity card with personal information and related articles.

4 Implementation

Based on the previous studies in entity search, we have combined techniques to improve our information retrieval process. The target was to build a search engine which would work for a multi-entity search. This approach is different in two aspects from previous works mentioned before in this thesis. 1) We are using domain-specific data which is structured around entities and 2) data sources, queries, and results are represented as entities. Instead of queries containing questions or sentences, we use entities as in term-based search, and the retrieved entities are returning a heterogeneous set of entities.

This chapter gives an overview of the design of the implementation and used technologies.

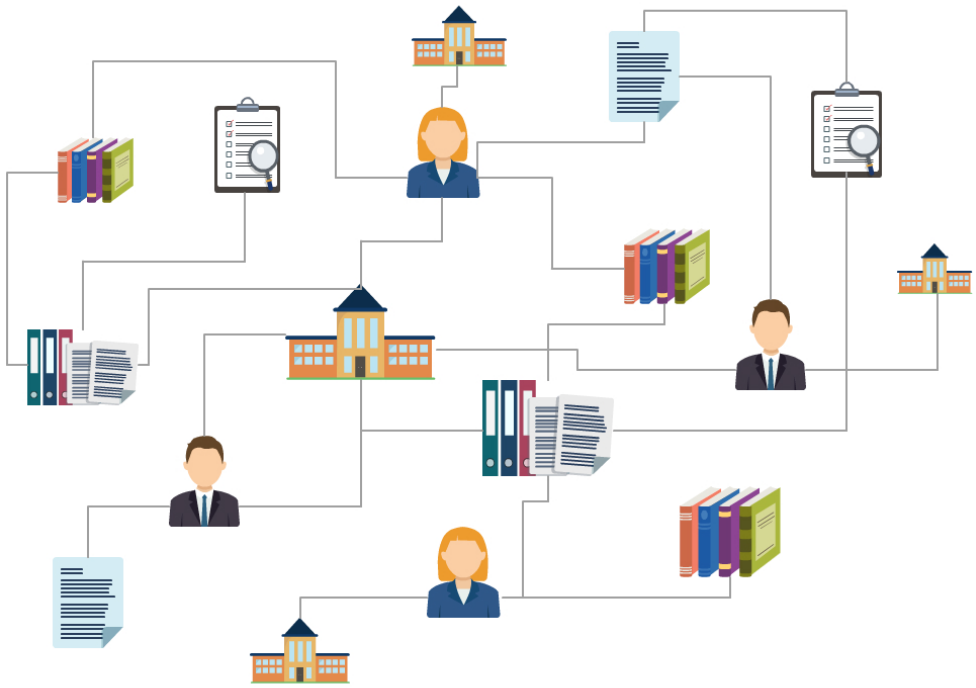

Figure 6: Linking all the entities creates a knowledge base

4.1 Data

The used data is from a research database of the University of Helsinki called TUHAT [4] portal, which contains different types of entities from persons to research outputs and events. After studying the data and its structure in TUHAT, the following was decided for this project: only 4 types of entities

[4]https://tuhat.helsinki.fi/portal/en/

are used: persons, organizational units, projects and research outputs, such as articles and patents. These types are stored to their own indexes for making efficient parallel query search possible. Each index has its own entity-specific field structure.

Each main entity has been stored as an entity card. Name and Universally Unique IDentifier (*UUID*) are the only attributes that are required to be in the entity card. Other attributes, like description, dates, related person, may or may not be presented. Some cases entity can have sub-entities which are not stored as their own units on the database. These kinds of sub-entities are e.g. keywords or external organizational units outside of the University of Helsinki. These entities can be collected through tags or UUIDs from the main entities, thus they don't need to be stored separately.

The total size of the database is 253534 entities.

The retrieved entities are returned as JSON objects. The following example presents the structure of a research output.

```
{
  "_type": "ro",
  "_id": "uuid",
  "_score": 1,
  "_source": {
    "author": [
      { "role": "Author", "type": "person", "uuid": "uuid", "name": "Name One" },
      { "role": "Author", "type": "person", "uuid": "uuid", "name": "Name Two" },
      {"..."}
    ],
    "abstract": "",
    "organisationalUnit": [
      { "type": "Department", "uuid": "uuid", "name": "Department 1" },
      {"..."}
    ],
    "publicationDate": {
      "year": "YYYY", "day": "DD", "month": "MM"
    },
    "keywords": [
      { "_type": "keyword", "_id": "Keyword1", "name": "Keyword 1" },
      {"..."}
    ],
    "type": "Conference contribution",
    "name": "Name of the research output",
    "metadata": {"..."}
  }
}
```

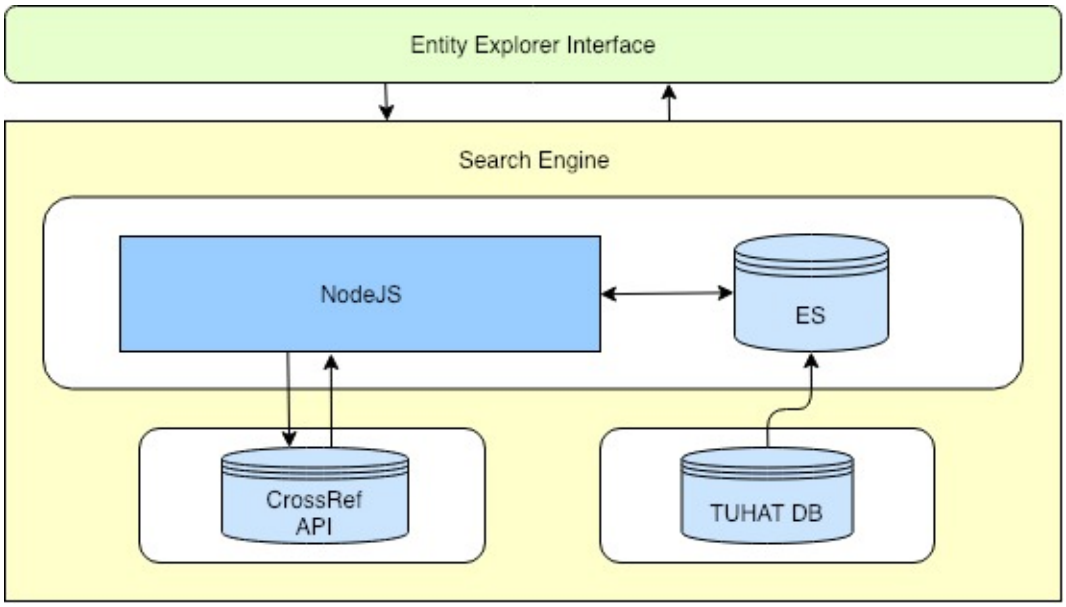

Figure 7: Overview of the system architecture.

4.2 System

In total three different platforms were used during the implementation. The implementation used **ElastichSearch** as the search engine and **NodeJS** as the API handler and contains the retrieval model. The last part was to retrieve metadata from **CrossRef API**. The overall structure of the system can be seen in Figure 7

Retrieval model is based on **the probabilistic model** and utilizes **query expansion**

4.3 ElastichSearch

ElasticSearch is a Java-based search engine, which is based on Lucene library [5]. The implementation was done using version 6.4 (*Elasticsearch Reference version 6.4*, 2018), which was the most recent version during the implementation period.

ElasticSearch was selected based on these factors: **the scalability**, **multitenancy**, **inverted indexing**, **ranking algorithm BM25** and it's easy management for indexing and searching (Hasibi et al., 2016).

[5]http://lucene.apache.org/

4.3.1 Okapi BM25

ElasticSearch has a built-in ranking function **Okapi BM25** (*Best Match*), which uses a probabilistic retrieval framework to rank matching documents based on the relevance to the given query (Robertson & Zaragoza, 2009; *Elasticsearch Reference version 6.4*, 2018). This approach can be used also in entity retrieval.

$$score_{BM25}(d,q) = \sum_{t \in q} log \frac{N}{N_t} \cdot TF_{BM25}(t,d) \tag{1}$$

Where $TF_{BM25}(t,d)$ is the inverse document frequency of the term

$$TF_{BM25}(t,d) = \frac{f_{t,d} \cdot (k_1 + 1)}{f_{t,d} + k_1 \cdot ((1-b) + b \cdot (l_d / l_{avg})} \tag{2}$$

Notation used in Eq. 1 and 2 (Büttcher et al., 2016)

d	is a document
q	is a query
N	is the number of documents in the collection
N_t	is the number of documents that contains the term t
t	is a term
$f_{t,d}$	is the number of occurrences of the term t within the document d
k_1	regulates the saturation of the TF
b	controls the degree of document length normalization
l_d	is the length of the document d, measured in tokens
l_{avg}	is the average length of all the collection's documents

ElasticSearch gives an opportunity to adjust the parameters k_1 and b.

4.4 Queries

The system has two kinds of query types for search: 1) free-form input search (as in Fiq. 1) and 2) multi-entity search based on entity UUID (as in Fiq. 4).

4.4.1 Single Entity Search

The query for an entity is based on free form input. This approach is similar to traditional term-based retrieval models (Hasibi et al., 2016; Sawant & Chakrabarti, 2013) The text input is the starting point for the retrieval. This input is passed on to the retrieval model which modifies it for ElasticSearch. The queries are processed, structured and weighted for each entity type separately.

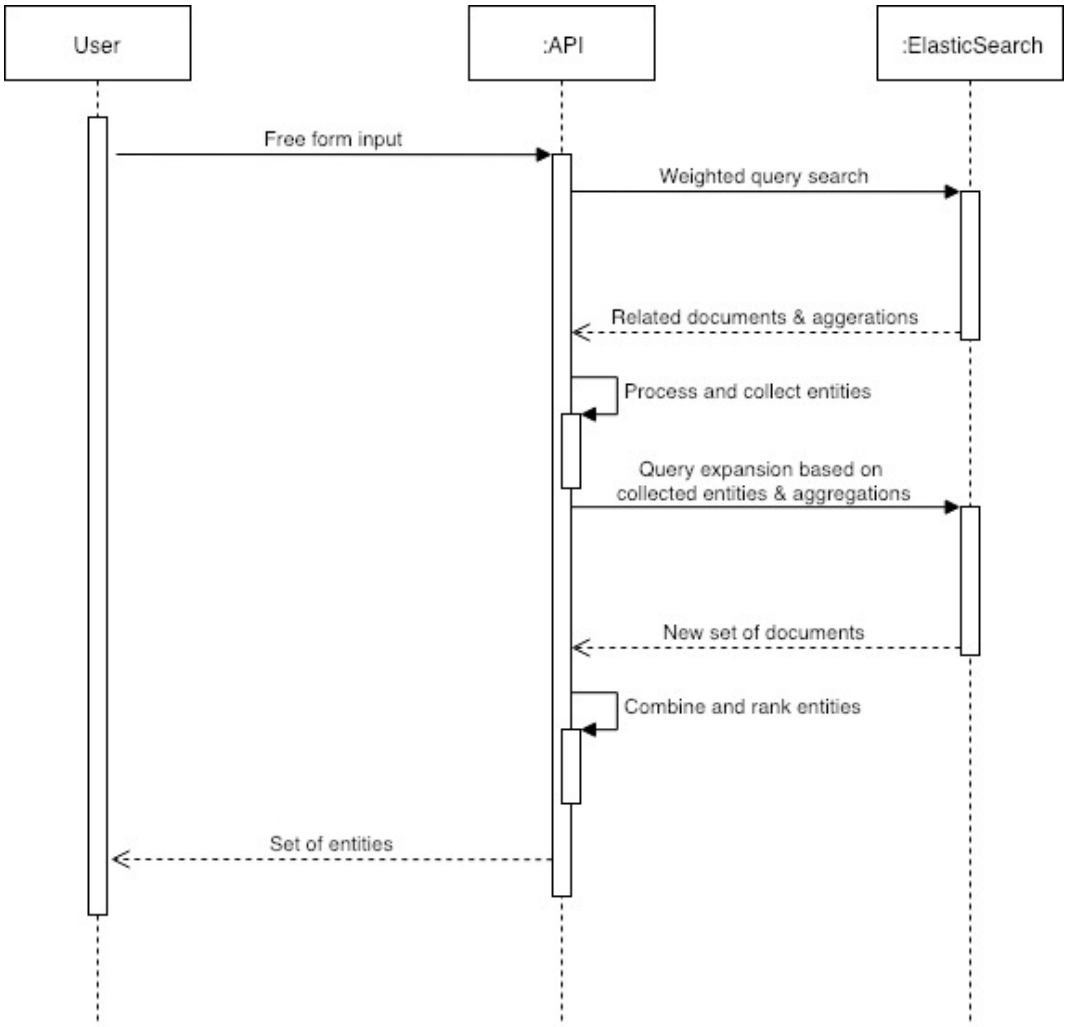

Figure 8: Sequence diagram of free format query search

For example, if the entity in search is a person, it is presumed that articles which in that person is the first author are more interesting comparing just for being part of it.

The top results from the initial search are then processed for the second search with query expansion. The query expansion is generated from different entities extracted from the results and combined with the top aggregations. All the results are looped through to extract keywords and top entities mentioned in the result sets. These keywords and top entities are stored and cross-referred to the second search.

The retrieval uses entity mention-level annotations (Balog, 2018). It can be used since we know that all the entities have their unique identifiers attached to them.

4.4.2 Multi-entity Search

Multi-entity search has a similar process as a single entity search (see Fig. 9), but the query is structured by using selected entities 1) type and 2) UUID. The idea of using only the identifiers for a more effective retrieval model came from the study of Hasibi et al. (2016), even though the implementation is different.

$$Query : [\{type_{e1}, uuid_{e1}\}, \{type_{e2}, uuid_{e2}\}...\{type_{en}, uuid_{en}\}]$$

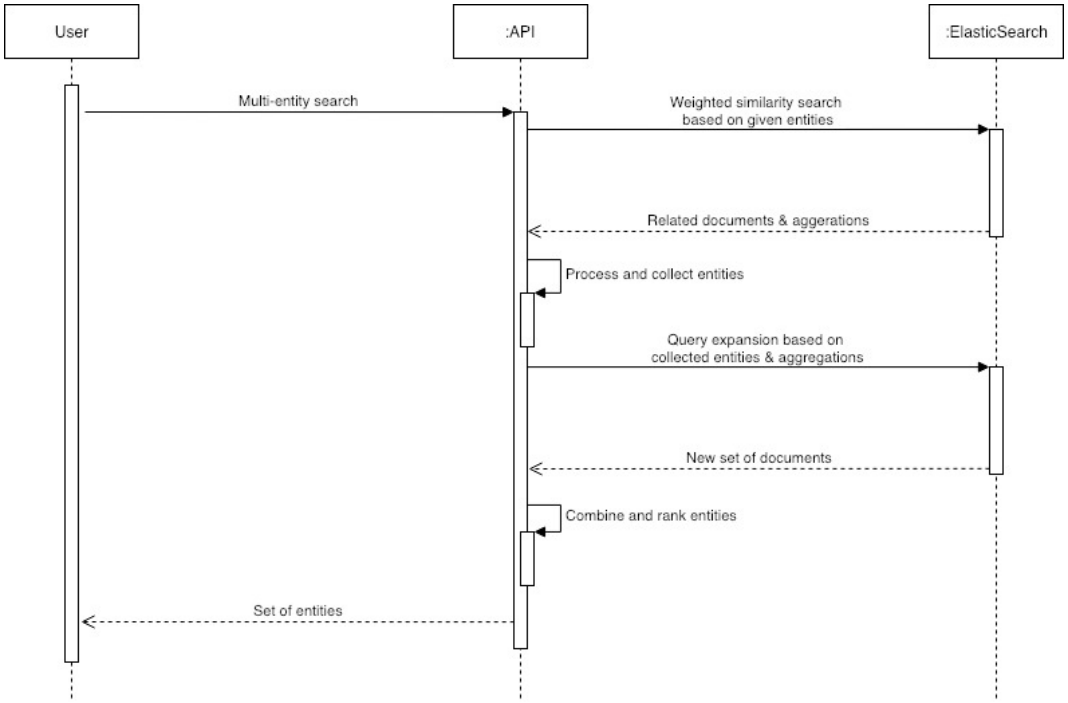

Figure 9: Sequence diagram of multi-entity search

4.4.3 Aggregation

Aggregations are part of the exploratory search. The purpose was to find decent alternative entities which might be interesting for the user. The aggregations are based on the search query and present summaries of the keywords and UUIDs that are in the result set (*Elasticsearch Reference version 6.4*, 2018). This means that aggregations can be used for the entity linking and find related entities for the searched entity. This will help the user to broaden the search and find connections.

The function to weight the aggregations for the query expansion was used in the following way:

$$f(n) = \begin{cases} k \cdot (1 + \frac{1}{k})^{\log_2 n} & \text{if type of } n \text{ is an UUID} \\ k \cdot (1 + \frac{1}{k})^{\log_{10} n} & \text{if type of } n \text{ is a keyword} \end{cases} \quad (3)$$

where n is a score of aggregation and k is presenting the times, it appeared in the initial search results. With logarithmic potency, the main weight is in the times it appeared in the search, which makes it a more valuable link to the wanted query but gives the opportunity to have other entities which are mentioned multiple times in the whole search set (aggregations).

4.4.4 Query Expansion

It's important to expand the query representation for avoiding a mismatch with explicit words and retrieving other associated entities (Dalton et al., 2014). The query expansion was built by merging the original query with aggregations and indirectly related entities extracted through entity linking from the initial result set (Liu et al., 2012). In this point, the initial term-based query is enriched with entity annotations. This has been proved to improve the performance of the retrieval and serving complementary information which is important to exploratory search.

4.4.5 Retrieving additional information

For the sake of fulfilling the user's information need, the system is fetching additional information from outside source CrossRef. Entity card is enriched with metadata (Fig. 10).

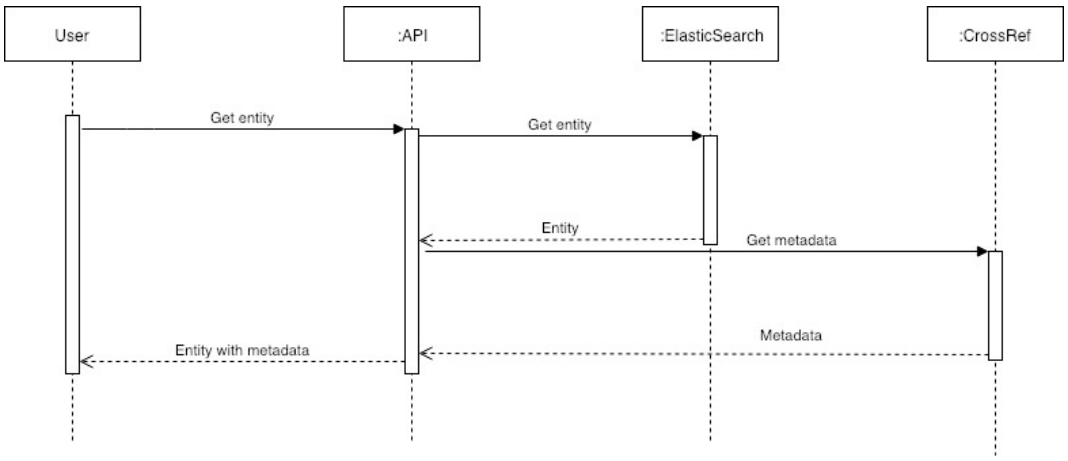

Figure 10: Fetching detailed information for a single entity based on UUID

4.4.6 Jaccard similarity

Jaccard similarity or *Jaccard similarity coefficient* (Shen et al., 2015) has been used in topical coherence between entities.

$$J(A, B) = \frac{|A| \cap |B|}{|A| \cup |B|} \tag{4}$$

Jaccard similarity was used in this implementation to calculate the similarity between A and B, where A is a combination of searched entities and their sub-entities, and B is an entity from the initial search results. Later, before the evaluation experiment, this adaption was discarded because of the uninteresting results it provided for the information need. This approach ranked entities with a less relevant value higher based on the sub-entities when we were looking for a higher ranking between topics e.g. distinguished researcher similar research areas versus research assistant working in the same research group. But this did not promote serendipity.

5 Evaluation

This chapter goes through the process and metrics which were used for evaluating the search engine described in the previous chapter. The evaluation in this study will focus on the effectiveness because we consider that the ElasticSearch has the efficiency to run queries fast and measuring this doesn't give any value to this study.

Relevance is measured through **precision** and **normalized discounted cumulative gain** that are widely studied and used metrics in information retrieval (Balog, 2018; Büttcher et al., 2016).

5.1 Experimental Design

We evaluate the entity-based retrieval system by comparing the implementation to a baseline. For the baseline, we used the state-of-the-art TUHAT word-based search engine. The data used in both systems are the same.

The experiment used within-subject design, where each participant performed all the tasks in both setups, in the implementation, and in the baseline.

5.2 Participants and Tasks

The evaluation was conducted with 13 users. All participants were students from the University of Helsinki and have prior-knowledge in academic search through research or thesis work. These participants also fit well to our domain of scientific information based on the TUHAT database. Each user had 4 types of tasks to perform (Table 2). Two of the tasks were single query searches, where the user used 1) a keyword and 2) a name of a researcher or a faculty member. The last 2 were multi-entity queries with 3) homogeneous and 4) heterogeneous entity types.

The results are based on the relevance feedback from those 13 users and were analyzed as the average from the outcomes in different query tasks.

Task	Task Definition
(1) Keyword search	Perform a search by using a keyword
(2) Person search	Perform a search by using a researcher's name
(3) Homogeneous entity search	Perform a search by using 2+ keywords
(4) Heterogeneous entity search	Perform a search by using keywords, names or any other entity types (2+)

Table 2: Tasks with definitions

5.3 Procedure and Set Up

The experiment was conducted through online calls, where users shared the desk view of their own laptops. First, they were asked to use a link, which directed them to a simple search engine of the implementation. After that the participants were asked to 1) find information regarding their thesis by using a keyword, 2) find information of a faculty member at University of Helsinki, 3) find more information related to their thesis by using multiple keywords and finally 4) find more information related to their thesis by using multiple keywords and names of a faculty member or an organization e.g. department.

After each task, users were asked to score the results one by one based on the relevance on the scale of 0 to 3, where 3 was the most relevant entity.

The same process was made with the baseline system, where users carried out the same kind of tasks (1-4) with evaluating them.

5.4 Evaluation Metrics

To see how well this implementation has achieved its intended purpose, it's needed to evaluate and measure by proper metrics. To assess the relevance of the results, we choose to evaluate the top 20 results retrieved for the queries the participants chose. Since the thesis focuses mainly on the retrieval model, we are measuring the retrieval performance by counting the precision and NDCG of the returned entities based on the queries user typed.

The following summary has the notations used on the metrics

Res	is the set of retrieved entities
Rel	is the set of relevant entities
rel_i	is relevance level

5.4.1 Precision

Precision (Büttcher et al., 2016; Balog, 2018) is the set of the retrieved entities, which is considered relevant to the given query. This is a standard metric for measuring effectiveness in IR. More precisely we are using Precision at k which is intended to measure effectiveness for ranked retrieval, considering our interest in the top 20 entities returned by the query.

$$P@k = \frac{|Res[1...k] \cap Rel|}{|Res|} \tag{5}$$

Precision is calculated for each task per participant separately and then combined those as an average for one task.

5.4.2 Normalized Discounted Cumulative Gain

In addition, **NDCG@20** (Balog, 2018; Büttcher et al., 2016) was also used to measure the effectiveness of the search engine. This method considers highly ranked entities on the result set to be more relevant to the user than the less relevant entities in lower ranks. This is to emphasize the variation of an information need and to address the quality of the result set. When evaluating the relevance between [0..3] for each top 20 entities, we can calculate:

$$NDCG(L) = \frac{DCG(L)}{IDCG} \tag{6}$$

NDCG is calculated from DCG *Discounted Cumulative Gain* and IDCG *Ideal Discounted Cumulative Gain*, where

$$DCG(L) = rel_1 + \sum_{i=2}^{n} \frac{rel_i}{\log_2 i} \tag{7}$$

and

$$IDCG = \sum_{i=1}^{|Rel|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{8}$$

This was chosen to evaluate the rank of the retrieved entities in a normalized way, where the size of the result set isn't affecting the outcome

5.5 Results

Metrics	Task 1		Task 2		Task 3		Task 4	
	I	B	I	B	I	B	I	B
P@5	0.93	0.87	1.00	0.87	0.93	0.47	0.90	0.47
P@10	0.90	0.67	0.97	0.81	0.93	0.27	0.90	NaN
P@20	0.85	0.58	0.93	0.76	0.90	NaN	0.95	NaN
NDCG	0.73	0.59	0.95	0.60	0.70	0.19	0.73	0.30

Table 3: General system performance categorized by the task. Precision at 5, 10 and 20, and Normalized Discounted Cumulative Gain measured from Our Implementation (I) and Baseline (B).

The table 3 shows the performance results. As seen from the table the overall performance is better in our implementation comparing the baseline. The biggest difference can be seen when considering top-20 results and multi-entity searches.

We also compared the retrieved result sets with the baseline and the implemented retrieval model. The results differed, on average, 0.30 per search,

which means that the implementation retrieved 30% more relevant entities than the baseline. This can be interpreted with the precision that we have managed to provide more relevant diverge results for the user and at the same time provide the key elements for facilitating exploratory search.

There are two downsides with our baseline. The search engine of the TUHAT portal doesn't provide keywords on the result hits. This might bias our results if we watch the results from task 4. With keywords, baseline might have had better or at least more retrieved entities. Secondly, the search isn't built for retrieving multiple entities at the same time. Nonetheless, with this, we can demonstrate that a multi-entity search has value for the user.

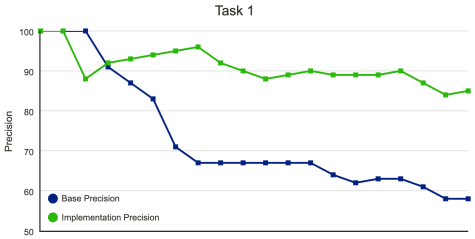

Figure 11: Precision plotted based on the rank for Task 1

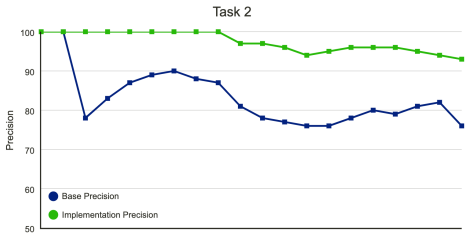

Figure 12: Precision plotted based on the rank for Task 2

Based on the results, our retrieval system with probabilistic model and query expansion, which utilizes aggregations and entity mentions, can retrieve relevant entities beyond the initial single entity search query. We can see from the plots 11 and 12 the decreasing of relevance in baseline and concurrently moderate decreasing with implementation line, which can be interpreted that users considered entities retrieved by our implementation relevant even at rank 20.

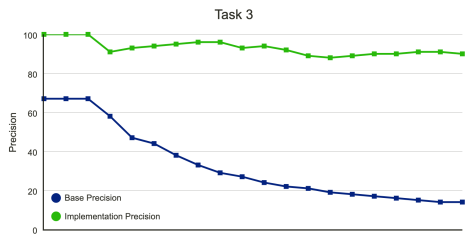

Figure 13: Precision plotted based on the rank for Task 3

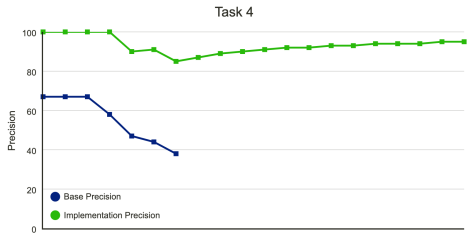

Figure 14: Precision plotted based on the rank for Task 4

As seen from the plot 13 and 14, queries for multi-entity search can improve information exploration. Especially in task 4, where the baseline could not offer enough relevant entities while it was possible for our retrieval model. This ensures more options for the user to browse in the information space and fulfill the information need.

RQ1 - *Does entity linking improve information exploration comparing to basic query search, especially when the search is made with multiple entities?*

Based on the results discussed before, we have managed to implement an improved retrieval system, which can handle queries with homogeneous or heterogeneous entity types simultaneously. The system can execute queries including 1 to 7 entities, hence we have accomplished the multi-entity search requirement.

Extending the probabilistic retrieval model with query expansion, which utilized entity linking, retrieved more extensive results sets, which participants still thought to be significant to their initial search queries. This can be demonstrated to be accurate from the results of NDCG.

RQ2 - *How to retrieve relevant and diverse results from different entity types?*

The starting point in this thesis was to find a way to utilize entities for a better retrieval model. The previous studies have shown that entity linking and query expansions have been used successfully to increase the relevance of the retrieved results. Hence, the implementation was designed by exploiting those.

Based on the experiment, using entity mentions and aggregations with the query expansion can lead to improved result sets, which increased the satisfaction of the participants for their information needs. The relevance of a result set with heterogeneous types was achieved by the system and based on the metrics, it was done without decreasing or compromising the relevance.

6 Discussion and Future Research

In this chapter, we go through our other findings besides the results from the experiment. The limitations which appeared during the implementation and evaluation are stated and the next steps for the future work are described.

Limitations

The data used in this study was restricted only for the data in the TUHAT database. It is unknown if this approach would work when exploiting open and wider data. Also, the entities were limited only for 4 types, whereas the TUHAT database has other types like events. It might provide more interesting information for the user if the rest of the entity types were included.

Another problem with the data was localization. Since the TUHAT database is owned by the University of Helsinki, there are cases where the articles, projects and other information are solely in Finnish. The lack of English entities led to missing relevant information and lowering the outcome in the search results.

In this experiment, the results are based on the feedback from users. To be noted that the users were students mostly majoring from computer science. This might affect the results in two different ways, 1) users might not be the best experts to evaluate the results from the search engine, although they are majoring in that area and 2) the richest data with English keywords and descriptions were crawled from the area of computer science and this leads to better hits for the search queries.

Another limitation regarding the experiment is that the order of the baseline and implementation was not counterbalanced. The experiment was always conducted so that all four tasks were done on implementation first and afterward the same four tasks were done on the baseline. This might have caused an order effect on the users, which can lead to biased results after the user learns the implemented system or gets bored after four tasks in a row with one system. This experiment should have been done in random order.

The lack of English localization and poor keyword usage had an impact on user satisfaction in other majors excluding generally computer science.

Future work

Next, we are mentioning ideas for future research or improvements, which were out of the scope in this study.

The solution could utilize lexicalization (alternative names, aliases, for entities and concepts) such as Wordnet[6], or other thesaurus libraries to provide higher quality in search results, especially during the keyword based single

[6]http://wordnet.princeton.edu/

query search.

Another improvement is related to metadata and CrossRef API. Not all the articles are added to their database, hence finding another source for metadata and adding that information to the ElasticSearch database might increase the relevance and interesting topics.

For wider infrastructure and through that providing more interesting data for the user, other knowledge bases and repositories in scientific domains could be crawled. This would, of course, transfer the main focus from the research made in the University of Helsinki but could provide other valuable knowledge.

Also, it might be considered for ensuring better quality, to make tagging compulsory for the researchers to add while creating new data entry. This helps not only improve the result hits but also to structure data more efficiently.

7 Conclusion

In this study, we have examined the use of entities for information retrieval system and the points of interest were how to design and implement entity-based search engine which also supports exploratory search, regardless of whether or not the search was made by using basic free form query or query with multiple entities. It is essential to improve search engines to support different types of information need in the incessantly expanding information space.

Based on the background research, we implemented this entity retrieval system with a probabilistic model using ElasticSearch and improving queries with entity linked query expansions. Query expansion was built from entity mentions and aggregations from the initial search result. Entity search can be exploited for exploratory search, but together with an entity retrieval system, visualization of the search engine has a key role in it.

For the first research question, *Does entity linking improve information exploration comparing to basic query search, especially when the search is made with multiple entities?* extending the probabilistic retrieval model with query expansion, which utilized entity linking, retrieved more extensive results sets, which users still thought to be significant to their initial query. This was seen especially in the multi-entity search comparing the baseline. In general, the implementation returned on average 30% more entities relevant to the search.

For the second research question, *How to retrieve relevant and diverse results from different entity types?* based on the results, the relevance of a result set with heterogeneous types was achieved by the system. At the same time novel and diverse information was provided to some extent. Adding entity mentions and aggregations for the initial query made results to be more diverse, without decreasing or compromising the relevance.

We have emphasized the importance of an information need and suggested a solution that could provide semantically, contextually and non-obviously related entities for vague search terms and queries. It still has improvements to be made, but based on the results of the current implementation, we have demonstrated that entity retrieval does improve the retrieved results from an aspect of relevance, novelty and diversity.

References

Athukorala, K., Głowacka, D., Jacucci, G., Oulasvirta, A., & Vreekenc, J. (2015, July). Is exploratory search different? a comparison of information search behavior for exploratory and lookup tasks. *Journal of the Association for Information Science and Technology*, *67*(11), 2635-2651. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.23617` (https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.23617)

Baeza-Yates, R. A., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Balog, K. (2018). *Entity-oriented search*. Springer International Publishing. doi: 10.1007/978-3-319-93935-3

Balog, K., Meij, E., & de Rijke, M. (2010). Entity search: Building bridges between two worlds. In *Proceedings of the 3rd international semantic search workshop* (p. 9:1-9:5). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/1863879.1863888` doi: 10.1145/1863879.1863888

Bordino, I., Mejova, Y., & Lalmas, M. (2013). Penguins in sweaters, or serendipitous entity search on user-generated content. In *Proceedings of the 22nd acm international conference on information & knowledge management* (pp. 109–118). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/2505515.2505680` doi: 10.1145/2505515.2505680

Bron, M., Balog, K., & de Rijke, M. (2010). Ranking related entities: Components and analyses. In *Proceedings of the 19th acm international conference on information and knowledge management* (p. 1079-1088). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/1871437.1871574` doi: 10.1145/1871437.1871574

Büttcher, S., Clarke, C., & Cormack, G. (2016). *Information retrieval: Implementing and evaluating search engines*. MIT Press.

Croft, B., Metzler, D., & Strohman, T. (2009). *Search engines: Information retrieval in practice* (1st ed.). USA: Addison-Wesley Publishing Company.

Dalton, J., Dietz, L., & Allan, J. (2014). Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international acm sigir conference on research & development in information retrieval* (pp. 365–374). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/2600428.2609628` doi: 10.1145/2600428.2609628

Elasticsearch Reference version 6.4. (2018). `https://www.elastic.co/guide/en/elasticsearch/reference/6.4/index.html`. ([Online; accessed 11-09-2018])

Gottipati, S., & Jiang, J. (2011). Linking entities to a knowledge base with query expansion. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 804–813). Stroudsburg, PA, USA:

Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=2145432.2145523

Hasibi, F., Balog, K., & Bratsberg, S. E. (2016). Exploiting entity linking in queries for entity retrieval. In *Proceedings of the 2016 acm international conference on the theory of information retrieval* (pp. 209–218). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2970398.2970406 doi: 10.1145/2970398.2970406

Kim, J. (2009, April). Describing and predicting information-seeking behavior on the web. *J. Am. Soc. Inf. Sci. Technol.*, *60*(4), 679–693. Retrieved from http://dx.doi.org/10.1002/asi.v60:4 doi: 10.1002/asi.v60:4

Klouche, K., Ruotsalo, T., Cabral, D., Andolina, S., Bellucci, A., & Jacucci, G. (2015). Designing for exploratory search on touch devices. In *Proceedings of the 33rd annual acm conference on human factors in computing systems* (pp. 4189–4198). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2702123.2702489 doi: 10.1145/2702123.2702489

Klouche, K., Ruotsalo, T., & Jacucci, G. (2018). From hyperlinks to hypercues: Entity-based affordances for fluid information exploration. In *Proceedings of the 2018 designing interactive systems conference* (pp. 401–411). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/3196709.3196775 doi: 10.1145/3196709.3196775

Klouche, K., Ruotsalo, T., Micallef, L., Andolina, S., & Jacucci, G. (2017). Visual re-ranking for multi-aspect information retrieval. In *Proceedings of the 2017 conference on conference human information interaction and retrieval* (pp. 57–66). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/3020165.3020174 doi: 10.1145/3020165.3020174

Li, X., Li, C., & Yu, C. (2010). Entityengine: Answering entity-relationship queries using shallow semantics. In *Proceedings of the 19th acm international conference on information and knowledge management* (p. 1925-1926). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1871437.1871766 doi: 10.1145/1871437.1871766

Liu, X., Fang, H., Chen, F., & Wang, M. (2012). Entity centric query expansion for enterprise search. In *Proceedings of the 21st acm international conference on information and knowledge management* (pp. 1955–1959). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2396761.2398551 doi: 10.1145/2396761.2398551

Metzger, S., Schenkel, R., & Sydow, M. (2013). Qbees: Query by entity examples. In *Proceedings of the 22nd acm international conference on information & knowledge management* (p. 1829-1832). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2505515.2507873 doi: 10.1145/2505515.2507873

Pérez-Agüera, J. R., Arroyo, J., Greenberg, J., Iglesias, J. P., & Fresno, V. (2010). Using bm25f for semantic search. In *Proceedings of the 3rd international semantic search workshop* (pp. 2:1–2:8). New York, NY,

USA: ACM. Retrieved from http://doi.acm.org/10.1145/1863879.1863881 doi: 10.1145/1863879.1863881

Robertson, S., & Zaragoza, H. (2009, April). The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, *3*(4), 333–389. Retrieved from http://dx.doi.org/10.1561/1500000019 doi: 10.1561/1500000019

Ruotsalo, T., Athukorala, K., G, D., Konyushkova, K., Oulasvirta, A., Kaipiainen, S., ... Jacucci, G. (2013). Supporting exploratory search tasks with interactive user modeling. In *Proceedings of the 76th asis&t annual meeting: Beyond the cloud: Rethinking information boundaries* (pp. 39:1–39:10). Silver Springs, MD, USA: American Society for Information Science. Retrieved from http://dl.acm.org/citation.cfm?id=2655780.2655819

Ruotsalo, T., Jacucci, G., Myllymäki, P., & Kaski, S. (2014, dec). Interactive intent modeling: Information discovery beyond search. *Commun.ACM*, *58*(1), 86-92. Retrieved from http://doi.acm.org/10.1145/2656334 doi: 10.1145/2656334

Salton, G. (1968). *Automatic information organization and retrieval.* McGraw Hill Text.

Sawant, U., & Chakrabarti, S. (2013). Learning joint query interpretation and response ranking. In *Proceedings of the 22nd international conference on world wide web* (p. 1099-1110). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2488388.2488484 doi: 10.1145/2488388.2488484

Shen, W., Wang, J., & Han, J. (2015, Feb). Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, *27*(2), 443-460. doi: 10.1109/TKDE.2014.2327028

Vallet, D., & Zaragoza, H. (2008). Inferring the most important types of a query: A semantic approach. In *Proceedings of the 31st annual international acm sigir conference on research and development in information retrieval* (p. 857-858). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1390334.1390541 doi: 10.1145/1390334.1390541

Xiong, C., Liu, Z., Callan, J., & Hovy, E. (2017). Jointsem: Combining query entity linking and entity based document ranking. In *Proceedings of the 2017 acm on conference on information and knowledge management* (p. 2391-2394). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/3132847.3133048 doi: 10.1145/3132847.3133048